ISSN 1561-5359. Штучний інтелект, 2018, № 4

УДК 004.491

О.С. Савенко
Хмельницький національний університет, Україна
вул. Інститутська, 11, м. Хмельницький, 29016

МЕТОД ВИЯВЛЕННЯ БОТ-МЕРЕЖ РОЗПОДІЛЕНИМИ СИСТЕМАМИ НА ОСНОВІ САМООРГАНІЗАЦІЇ

O.S. Savenko
Khmelnytsky National University, Ukraine
11, Instytutska St., Khmelnytsky, 29016

THE METHOD OF DETECTION OF BOT-NET ON BASED DISTRIBUTED AND SELF-ORGANIZATION SYSTEMS

Розроблено метод виявлення бот-мереж розподіленими системами на основі самоорганізації та метод взаємодії її компонентів. Розроблені методи підтримують самоорганізацію розподіленої системи, самостійність у прийнятті рішень компонентами системи без стороннього впливу, організацію взаємодії компонентів для визначення стратегії подальшої роботи, прийняття рішень про кількість компонентів системи, прийняття рішень про наявність зловмисного програмного забезпечення у певних комп'ютерних системах через залучення інших компонентів розподіленої системи з наступним відключенням таких комп'ютерних систем, обробку і передачу знань про нове виявлене зловмисне програмне забезпечення від одних компонентів системи іншим, оцінку стану безпеки окремих компонентів системи та її в цілому.

Ключові слова: зловмисне програмне забезпечення, розподілена самоорганізаційна система, локальна мережа

The method of detection of botnets by distributed systems on the basis of self-organization and the method of interaction of its components is developed. The developed methods support the self-organization of the distributed system, the autonomy in decision making by the components of the system without third-party influence, the organization of the interaction of the components to determine the strategy of further work, decision-making on the number of components of the system, making decisions about the presence of malware in certain computer systems through the involvement of other components distributed system with the subsequent disconnection of such computer systems, processing and transferring knowledge about the new detected malice not software from one component of the system to another, an assessment of the security status of individual components of the system and its overall.

Keywords: malware, distributed self-organizing system, local area network

Вступ

Невпинне щоденне збільшення та використання зловмисного програмного забезпечення (ЗПЗ) створює проблеми користувачам комп'ютерних систем (КС). Отримання фінансової вигоди або іншої переваги вмотивовують розробників зловмисного програмного забезпечення до його збільшення та розповсюдження [1, 2]. Ними до його створення залучаються сучасні технології розробки програмних засобів, у тому числі і розподілених. На сьогодні сучасне ЗПЗ може бути розроблене у вигляді складних багатофункціональних програмних систем та комплексів, які побудовані з використанням ефективних методів поширення зловмисного коду та подальшого адміністрування його роботи. Особливо актуальним на сьогодні є виявлення ЗПЗ, яке поширюється комп'ютерними мережами. Наявні засоби його виявлення на сьогодні не задовольняють потреб користувачів. Особливо це стосується задач по виявленню ЗПЗ на випередження, на етапі його початкового поширення. Відомі різноманітні антивірусні засоби, які здійснюють виявлення ЗПЗ на різних етапах його функціонування, не забезпечують повного його виявлення [1, 2]. Для виявлення у КС локальних мереж використовуються мережні антивірусні програмні засоби (АПЗ) [3-7]. Вони дозволяють організувати процес виявлення у сукупній кількості КС за рахунок більшої обчислювальної потужності порівняно з

окремими КС. Переважно такі засоби використовують у корпоративних мережах організацій та підприємств. Основною характерною особливістю існуючих мережних АПЗ є наявність централізованої архітектури, що може бути використана зловмисниками для їх блокування, після виявлення центрів. Тому, проблема розробки нових методів та систем виявлення ЗПЗ для застосування в локальних мережах залишається актуальною.

Постановка проблеми

Ефективна протидія мережному ЗПЗ, зокрема бот-мережам, може бути здійснена мережним АПЗ, архітектура яких була б подібною до архітектури складного комплексу ЗПЗ, але мала б більшу функціональність. Досягнення підвищення достовірності виявлення ЗПЗ у межах тільки однієї комп'ютерної системи, яка має вихід у мережу Internet, може бути недостатнім при протидії засобам ЗПЗ, які представлені великим програмним комплексом, що розміщений у багатьох комп'ютерних системах у глобальній мережі. Тому, підвищення достовірності виявлення ЗПЗ можливе за рахунок залучення груп КС порівняно з однією КС. Такими групами КС можуть виступати КС локальних комп'ютерних мереж (ЛКМ). Відповідно важливим завданням є здійснення побудови таких ефективних систем для їх використання в локальних комп'ютерних мережах, де уможливлюється встановлення таких систем на усіх комп'ютерних системах мережі, а не локально. Це дозволить при проведенні аналізу зібраних типових даних з різних КС та їх відповідної обробки підвищити достовірність виявлення.

Ефективне застосування методів і засобів виявлення ЗПЗ потребує розробки системи, яка б включала в себе достатню кількість реалізованих ефективних методів у вигляді відповідних підсистем, мала можливість до нарощування та враховувала б майбутні тенденції розвитку як антивірусних засобів, так і ЗПЗ. Оскільки, бот-мережі є керованим розподіленим програмним забезпеченням зловмисника, то перспективним напрямом досліджень на противагу є розроблення теорії і практики створення розподілених систем виявлення [8-11]. Важливими задачами при цьому є розробка методу виявлення бот-мереж із застосуванням розподілених самоорганізованих систем [11] та методу взаємодії їх компонентів у локальних мережах, особливістю якого була б можливість такої самоорганізації системи, що надавала б узгоджену підтримку методам виявлення безпосередньо мережного ЗПЗ.

Аналіз останніх досліджень і публікацій

Відомі методи (метод сигнатурного аналізу, метод контрольних сум, метод евристичного аналізу та інші) виявлення ЗПЗ переважно орієнтовані на застосування в кінцевих КС. Для антивірусних засобів мережного типу розроблено методи, застосування яких є можливим переважно на сервері або в корпоративних чи локальних мережах. Більшість із цих методів розроблено з використанням технологій та компонентів штучного інтелекту. Як правило, сучасні системи виявлення ЗПЗ містять набори багатьох методів та їх комбінацій, на що впливає зростання різновидів ЗПЗ. Розглянемо детальніше відомі системи та методи для виявлення ЗПЗ.

Авторами роботи [12, 13] запропоновано систему ідентифікації та класифікації для мережних кібератак. Для реалізації системи запропоновано використання комбінації різних методів штучного машинного навчання, а саме: нейронних мереж, імунної системи, нейрофізичних класифікаторів та метод опорних векторів. Відмінною особливістю запропонованої системи є багаторівневий аналіз мережного трафіка, що дає можливість виявляти атаки методом підпису та комбінувати набір адаптивних детекторів на основі методів машинного навчання.

У роботі [14] запропоновано статичну систему виявлення ЗПЗ, що заснована на використанні методу головних компо-

нентів для вилучення даних та класифікаторів SVM, J48 та NaiveBayes для формування висновку. Для усунення недоліків відомих антивірусних засобів залучаються методи статичного аналізу для формування ознак, отриманих з інформації про Windows PE заголовок, DLL бібліотеки та API виклики всередині кожної DLL бібліотеки. Для зменшення отриманої множини ознак використовується метод головних компонент.

Систему виявлення кібератак на основі залучення нейромережних імунних детекторів представлено у роботі [15]. Розроблена система складається з двох частин. Перша реалізована апаратно й працює постійно в режимі реального часу. Друга частина представлена програмним забезпеченням на виділеному комп'ютері, який використовується для аналізу поточних атак та створення відповідних засобів захисту. Прийняття рішення про можливий вплив ШПЗ здійснюється із залученням системи нейромережних детекторів, в основу якої закладено алгоритм Мамдані.

Іншим підходом до виявлення зловмисного програмного забезпечення є вирокремлення характеристичних ознак на основі інформації про потік виконання програми. У роботі [16] запропоновано систему, що передбачає побудову графу потоку керування зловмисної програми, з подальшою конвертацією його у векторний простір.

Окрім формування графу потоку керування для виявлення метаморфних вірусів застосовуються ознаки, сформовані на основі відстеження API викликів, що здійснює програма [11]. У роботі автори запропонували статичний підхід формування сигнатури метаморфного вірусу, що заснований на підрахунку кількості відповідних API викликів. Висновок про наявність метаморфного вірусу формується на основі пошуку схожості сформованої сигнатури з базою сигнатур з використанням метрик подібності.

Проведений аналіз показав, що для виявлення ЗПЗ відомі системи здійснюють аналіз мережного трафіка, файлів аудиту, пакетів, що передаються по мережі, перевіряють конфігурацію відкритих мережних сервісів. Для встановлення факту порушення роботи КС використовуються різні методи машинного навчання, а саме: нейронні мережі, штучні імунні системи, метод опорних векторів, Баєсові мережі, нечітку кластеризацію [17-19]. Основним недоліком відомих систем є їх хост-орієнтований підхід до виявлення ЗПЗ.

Для того щоб ефективно застосовувати методи та засоби виявлення ЗПЗ необхідно розробити систему, яка б включала в себе достатню кількість реалізованих ефективних методів у вигляді відповідних підсистем, мала можливість до нарощування та враховувала б майбутні тенденції розвитку як антивірусних засобів, так і ЗПЗ. Крім того, враховуючи вплив людського фактору при поширенні ЗПЗ, необхідним є побудова системи виявлення на такій архітектурі, яка б не залежала від його впливу. Тому, враховуючи розподіленість системи в локальних мережах важливою вимогою до неї є її самоорганізованість безпосередньо в роботі і при прийнятті рішення з виявлення ЗПЗ.

Мета дослідження

Метою дослідження є розробка методу виявлення бот-мереж розподіленими системами на основі самоорганізації та методу взаємодії її компонентів. Архітектура такої розподіленої багаторівневої системи (РБС) представлена в [11], яка також, на відміну від існуючих мережних АПЗ [3-7], є децентралізованою. Врахування особливостей архітектури розподіленої системи, яка є самоорганізованою, надає переваги при виявленні ЗПЗ. Такими перевагами є самостійність у прийнятті рішень без стороннього впливу (наприклад, адміністратора мережі), організація взаємодії компонентів РБС для визначення стратегії подальшої роботи, прийняття рішень про

кількість компонентів системи динамічно, прийняття рішень про наявність ЗПЗ у певних КС через залучення інших компонентів РБС, з наступним відключенням таких КС, обробка і передача знань про нове виявлене ЗПЗ від одних компонентів системи іншим, оцінка стану безпеки окремих компонентів РБС та її в цілому.

Виклад основного матеріалу

Метод виявлення бот-мереж розподіленими системами на основі самоорганізації

Особливістю використання РБС виявлення ЗПЗ у локальних мережах є те, що з метою здійснення виявлення розроблені відомі методи, орієнтовані саме на виявлення ЗПЗ в окремих КС чи корпоративних мережах за певним його типом чи типами, але при цьому не враховують при виявленні особливості антивірусних засобів, у яких вони реалізовані. РБС розроблена для здійснення тривалого спостереження за протіканням процесів в окремих КС та локальній мережі. Це надає можливість накопичувати інформацію про процеси, які протікають у конкретній КС, порівнювати їх між різними КС локальної мережі. Також, перевагою РБС над ЗПЗ є те, що РБС містить у своїх компонентах інформацію про кожну КС локальної мережі і використовує її для порівняння в процесі свого функціонування. Оскільки місцем дослідження виступають КС локальної мережі, де адміністратор мережі встановив визначене загальне і спеціалізоване програмне забезпечення, тоді РБС отримує перевагу над ЗПЗ, якому необхідно буде пробувати отримувати інформацію про нього. Користувачі в локальній мережі обмежені у користуванні КС, але вони теж можуть несанкціоновано встановлювати потрібне їм програмне забезпечення, відвідувати веб-сайти із ЗПЗ. Враховуючи, що вузли бот-мережі в КС можуть не проявлятись швидко, перебуваючи в стані очікування і збору потрібної інформації або тримаючи під контролем КС не здійснювати інших дій, то їх виявлення на основі відслідковування можливих зловмисних дій потребує не тільки тривалого спостереження, що є важливим, але і відповідних методів виявлення та структури засобів виявлення, які б надавали суттєві переваги при виявленні. Методи виявлення ЗПЗ повинні враховувати особливості засобів, у яких вони будуть реалізовані. Їх реалізація без отримання переваги за рахунок структури засобів не зможе надати їм переваги особливо тоді, коли такі засоби будуть встановлюватись у КС, у якій вже присутнє ЗПЗ.

При розгляді сукупності КС у локальній мережі припустимо, що частина з них є вузлами бот-мережі, а решта не є частиною бот-мережі. При цьому можуть бути варіанти, при яких КС можуть містити файлове ЗПЗ, яке не відноситься до бот-мереж. А також, з КС може проводитись атака на іншу КС або здійснюватимуться зловмисні дії в КС. Якщо КС містить вузол бот-мережі чи файлове ЗПЗ, то на КС може здійснюватись атака ззовні з вузлів іншої бот-мережі чи проводитись іншим мережним ЗПЗ. Врахуємо також, що при відсутності мережного ЗПЗ у КС атака з цієї КС не відбувається. Крім того, вважатимемо, що усі розглянуті КС містять компоненти РБС, тобто в них здійснюється моніторинг подій і приймається рішення про виявлення мереж-ного ЗПЗ, у тому числі вузлів бот-мереж. Виділимо та представимо ці 12 варіантів у табл. 1 згруповано, залежно від наявності бот-мереж у КС:

1) вузол бот-мережі в КС здійснює зловмисні дії тільки в межах КС і атака з КС на інші КС не проводиться, але здійснюється або не здійснюється атака на цю ж розглядувану КС іншим мережним ЗПЗ, при цьому інше ЗПЗ у КС може або не може бути присутнім;

2) те ж що і у варіанті 1, тільки з КС здійснюється атака вузлом бот-мережі;

3) вузла бот-мережі в КС немає, наявне файлове ЗПЗ, атака на КС ззовні здійснюється або не здійснюється;

4) ЗПЗ немає, атака ззовні на КС здійснюється або не здійснюється.

Таблиця 1. Варіанти подій у КС

№ варіанту подій	Вузол бот-мережі в КС	Файлове ЗПЗ в КС	Атака з КС	Атака на КС
1	+	-	+	+
2	+	-	+	-
3	+	-	-	+
4	+	-	-	-
5	+	+	+	+
6	+	+	+	-
7	+	+	-	+
8	+	+	-	-
9	-	+	-	+
10	-	+	-	-
11	-	-	-	+
12	-	-	-	-

Події, які не представлені в табл.1, не можуть відбуватись. Зокрема, якщо немає ЗПЗ в КС, тоді з неї не може вестись атака на інші КС. Варіанти подій з табл.1 є основою для розробки методу виявлення бот-мереж у частині відображення в ньому особливостей різного стану КС, у якому вона може перебувати. Для здійснення обробки подій у КС компонентою РБС необхідно здійснити формалізоване представлення розроблених функцій бот-мереж через поведінкові сигнатури на основі API функцій. Підготовка таких еталонних моделей полягає у формуванні знань на основі функцій, які вказують на наявність бот-мереж. Отримавши вектори, в яких представлені числові величини як їх компоненти, що дозволяє на основі їх перейти до здійснення класифікації нових виконуваних об’єктів у КС, потрібно організувати збір інформації про виконувані процеси в КС на основі API функцій та розробити метод їх обробки для перевірки віднесення до одного з класів бот-мереж.

Проаналізуємо можливі варіанти подій у КС за присутності в ній компоненти РБС та ЗПЗ. На основі їх дослідження визначимо стратегії поведінки ПМ у процесі появи таких подій.

Розглянемо випадок, коли в КС вже створено вузол бот-мережі, тобто завантажено ЗПЗ вузла бот-мережі, отримано контроль над певним програмним забезпеченням КС. Якщо вузол бот-мережі міститься в КС, тоді при її запуску стартове зловмисне навантаження повинно проявити себе в одному з процесів, які створюються прописаними виконуваними програмами в файлах автозапуску. Інакше воно не зможе активуватись у КС при кожному її увімкнені і, як наслідок, буде вилучено з часом з бот-мережі. Або воно очікуватиме запуску користувачем певних програм чи програми, що теж може відбутись або не відбутись. Тому, висуваємо гіпотезу, що запуск програмного забезпечення вузла бот-мережі здійснюється після увімкнення КС. У процесі отримання контролю над КС бот-мережа чи на перших етапах свого функціонування вузол бот-мережі здійснить призупинення роботи відомих анти-вірусних засобів та перейде в режим імітації їх роботи.

Варіантів встановлення компонент РБС може бути два: 1) під час нового встановлення програмного забезпечення (ПЗ) КС встановлюється програмне забезпечення компоненти; 2) програмне забезпечення компоненти встановлюється у вже функціонуючу КС. При першому варіанті КС може бути новою і тому потребує встановлення ПЗ або вже використовуваною раніше і потребує повного переустановлення ПЗ. Але навіть, якщо вона є новою, то, як правило, містить операційну систему і до включення її до складу локальної мережі вже міститиме певне визначене ПЗ. Таким чином, вірогідність отримати ЗПЗ при першому варіанті дуже мала, але теоретично можлива. При другому варіанті ймовірність наявності ЗПЗ в КС більша, ніж в першому.

Якщо припустити, що ЗПЗ у першому варіанті не було, тоді компонента РБС встановлюється першою і отримує доступ для контролю над всією КС. Цей контроль передбачає такі основні дії: запуск першої, звірку виконуваних програм з файлу автозапуску, розміщення резидентної програми в пам’яті, моніторинг API викликів і збір їх у вектори.

Програмне забезпечення вузла бот-мережі може потрапити в КС двома способами: користувач, мережа. Тоді таке ЗПЗ буде обов'язково пробувати прописати себе для можливості активації при наступному ввімкнені КС, створювати можливості для розміщення своєї резидентної частини в пам'яті. При спробі прописатись у компоненті РБС буде виявлено через здійснення самоконтролю при запуску, який закладено в функціонал компоненти, якщо ЗПЗ допустить запуск наступних після нього команд. Якщо ж ЗПЗ не допустить запуск команд компоненти, тоді він не відзвітується перед рештою компонент РБС і буде ними вилучений із РБС, що дозволить блокувати таку КС. При другому варіанті: якщо ЗПЗ не було, а з'явилось, тоді та ж стратегія, що і для першого розглянутого варіанту. А якщо ЗПЗ вже було, тоді в процесі функціонування КС виникне конфлікт між ПЗ вузла бот-мережі і компонентою, суть якої полягатиме в змаганні за перший запуск та доступ до оперативної пам'яті, облік для компоненти і блокування відомих АПЗ. Певний час вони можуть функціонувати. У будь-якому з варіантів компонента виступить об'єктом для атаки як приманка. Але не все ЗПЗ, а особливо бот-мережі, розроблено на основі стратегії змагання за повний контроль над КС чи перший запуск. Для приховування своєї присутності стратегія перебування в КС може передбачати, наприклад, тільки прописування в одному з файлів, які містяться у файлі автозапуску. Тоді виявити таке ЗПЗ можна лише за його проявами, важливим з яких для вузла бот-мережі є необхідність підтримки зв'язку зі своїм контролюючим центром. Тобто, для цього випадку вузол бот-мережі не проявляє активності, а очікує команди і потім запускає повний пакет програмного забезпечення, необхідний для її виконання. Така стратегія дозволяє перебувати в КС тривалий час без виявлення. Але ці та інші стратегії, закладені в механізм функціонування бот-мережі, можуть порушити інші події, які викликані сторонніми проявами. Наприклад, розглянемо перший варіант подій з табл.1, тобто, коли отримано команду здійснити атаку на іншу КС чи ресурс, а в цей час подібна атака відбувається на цю ж КС. Така подія може відбуватись тільки тоді, коли атака на КС ведеться не з цієї ж бот-мережі, вузол якої міститься в КС, а іншим ЗПЗ. У цьому випадку компонента РБС переходить до стану 7 [11], який активується через надмірне звернення до портів та викликом функцій, орієнтованих на встановлення мережних з'єднань. Компонента визначає IP адресу, куди націлена інтенсивна відправка пакетів, і здійснює збільшення інтервалу надсилання пакетів через блокування відповідних пакетів. Проведення атаки на цю ж КС, яка є в локальній мережі, не може містити ресурсів для атаки, і може здійснюватись тільки іншим зловмисником для встановлення контролю над нею. Ця атака може відбуватись з іншої КС цієї ж мережі, тоді компонента визначає КС, про яку повідомляє решті компонентів РБС для здійснення її дослідження, або через межу локальної мережі. У будь-якому з випадків відбивання атаки здійснюватиметься відповідними засобами. Але частина атакуючих дій може бути успішною, тоді в КС будуть одночасно присутні компоненти, ПЗ вузли бот-мережі та нове ЗПЗ. При їх функціонуванні виникне конфлікт, який проявиться у спробі здійснення контролю над КС, що буде впливати на її нормальний порядок функціонування. У цьому випадку компонента, при звірці інформації про розміщені в КС файли, виявить поточні зміни і почне дослідження підозрілих файлів.

У другому варіанті, якщо в КС міститься вузол бот-мережі і з неї здійснюється атака, тоді компонента порівнює зростаючу інтенсивність викликів мережних з'єднань і здійснює їх затримки методом призупинення та виявляє процес, який їх генерує. Цей варіант можливий при повному контролі компоненти в КС і другорядній ролі вузла бот-мережі, якщо ж інакше, то тобі

вузол бот-мережі змагався б за ресурси КС з компоненти і цим виявив би себе.

У третьому варіанті подій розглядаємо можливість проникнення в КС з малою ймовірністю. При цьому компонента аналізуватиме інтенсивність звернення до портів та надходження пакетів, а також подальшого розміщення файлів, що надійшли, і фіксування місця їх розміщення. У подальшому компонента розміщує такі файли в свій реєстр для спостереження за ними протягом певного часу.

У четвертому варіанті подій компонента здійснює змагання за ресурси з ПЗ вузла бот-мережі. І аналогічно, як у першому варіанті, може бути дві варіації: ПЗ вузла бот-мережі прагнутиме встановити повний контроль над КС або згідно зі своєю стратегією функціонування приховуватиме свою присутність до отримання команди на проведення атаки.

П’ятий варіант подій включає результати першого з врахуванням того, що додатково наявне в КС файлове ЗПЗ. Його присутність ускладнюватиме функціонування КС, бо воно перебуватиме в оперативній пам’яті, здійснюватиме своє поширення, шукаючи об’єкти для втілення. Характерною особливістю в цьому варіанті буде завантаженість ресурсів і сповільнення роботи КС. Компонента відмічатиме зміни в файлах, які прописані в її реєстрі для цієї КС. Основним місцем, де зіткнуться компонента РБС, ПЗ вузла бот-мережі та файлове ЗПЗ буде оперативна пам’ять. Атака на цю КС підсилюватиме ускладнення роботи КС, що аналогічно до першого варіанту призведе до тривалої обробки результатів моніторингу компоненти в КС і повідомлення про події іншим компонентам РБС.

Аналогічно до першого і п’ятого варіантів будуємо стратегії розвитку подій для шостого, сьомого та восьмого варіантів подій. Варіанти 9-12 стосуються файлового ЗПЗ, бо вони не передбачають наявність ПЗ вузла бот-мережі.

Стратегії компонент РБС в різних варіантах подій:

1) КС контролюється бот-мережею, компонента в КС заблокована вузлом бот-мережі, КС виконує поставлені кори-стувачем задачі;
2) КС контролюється бот-мережею, компоненту РБС в КС заблоковано вуз-лом бот-мережі, КС функціонує з трива-лими перебоями та затримками у вико-нанні запитів користувача;
3) КС контролюється бот-мережею, компонента в КС заблокована вузлом бот-мережі, КС функціонує з тривалими перебоями та затримками у виконанні запитів користувача, файлове ЗПЗ в КС здійснює своє поширення;
4) КС контролюється компонентою РБС, вузол бот-мережі в КС дослід-жується компонентою, КС виконує поставлені користувачем задачі;
5) КС контролюється компонентою, вузол бот-мережі та файлове ЗПЗ в КС досліджуються компонентою РБС, КС виконує поставлені користувачем задачі;
6) КС контролюється компонентою, яка здійснює моніторинг і обробку подій.

Проаналізуємо логіку взаємодії компоненти РБС та ПЗ вузла бот-мережі в КС для отримання стратегій. Задамо стадії функціонування варіантів подій і можливих стратегій їх розвитку часовою діаграмою та виділимо в ній повторювані фрагменти для здійснення оптимізації при прийнятті рішення компонентою РБС. Шаблони можливих варіантів подій у КС та їх варіації формують одну з шести стратегій. Стратегії для подій у КС локальної мережі представлено в табл. 2.

Метод виявлення бот-мереж у КС локальних мереж складається з таких основних кроків:

1. Отримання даних про активні процеси та мережні пакети на основі активного моніторингу виконання команд у КС (починаючи з першої API функції кожного процесу, що буде виконуватись після запуску КС).

Таблиця 2. Фрагмент стратегій для подій у КС локальній мережі

№ з/п	1		2		3	4	5	6	7	8
	К	Б	К	Б						
1	1	0	0	1	1	1	1	1	1	1
2	1	0	0	1	1	1	1	1	0	1
…	..	..	…	..	..	..	..	..	..	..
512	0	0	0	1	0	0	0	0	0	6

Позначення в табл. 2: 1 – встановлення в КС до (1)/після(0) компоненти РБС або ПЗ вузла бот-мережі; 2 – стартовий контроль у КС: так(1)/ні(1); 3 – атака з цієї КС: так(1)/ні(1); 4 – атака з цієї КС на КС цієї ж мережі: так(1)/ні(1); 5 – атака на КС: так(1)/ні(1); 6 – атака з КС цієї ж мережі: так(1)/ні(1); 7 – наявність файлового ЗПЗ, встановленого до(1)/після(0) встановлення компоненти РБС; 8 – стратегії; К – компонента РБС; Б – ПЗ вузла бот-мережі.

2. Здійснення збору даних моніторингу після виявлення певних імовірно зловмисних проявів у КС у вектор.
3. Формування вектору ознак імовірно підозрілих дій для зібраних даних, компонентами якого є API функції.
4. Прийняття рішення про місце обробки вектору ймовірно зловмисних дій.
5. Якщо аналіз завантаженості ресурсів КС показав невеликий відсоток завантаженості, тоді здійснити обробку в цій КС, інакше надіслати в іншу визначену компоненту КС.
6. Здійснення класифікації вектору імовірно зловмисних дій.
7. Аналіз результатів кроку 6.
   7.1. Якщо встановлено віднесення такого вектору до певного підкласу класу бот-мереж, тоді додавання цієї інформації до класифікаторів усіх компонент.
   7.2. Якщо встановлено віднесення такого вектору до декількох підкласів класів бот-мереж, тоді здійснити аналіз із залученням решти компонент РБС на основі обробки варіантів подій з табл. 1.
   7.3. Якщо близькість для включення до певного підкласу є нечіткою, але додатково із залученням решти компонент визначено, що вектор містить зловмисні дії, тоді здійснити створення нового класу для бот-мереж, занести дані, оновити налаштування класифікатора, передати результат решті компонент РБС.
   7.4. Якщо перевірка встановила, що досліджуваний вектор не містить зловмисного навантаження, тоді здійснити зупинку дослідження процесу, на основі якого він був сформований.
   7.5. Якщо встановлено, що досліджуваний вектор містить зловмисне навантаження, тоді здійснити зупинку відповідного процесу.
   7.6. Здійснення пошуку і дослідження на основі отриманих відомостей аналогічних процесів в інших КС мережах, де встановлена РБС, її компонентами.
8. Обробка варіантів з табл. 1 та табл. 2 із залученням решти компонентів РБС. На основі варіантів подій табл. 1 та табл. 2 виокремлення варіантів, у яких можливе відключення ЗПЗ компонентів РБС, і встановлення такої події для оцінки іншими компонентами РБС. У цьому випадку здійснюється вилучення компоненти з РБС.
   8.1. Для варіантів 1-256 задіяти стратегію 1 на основі прийняття рішення рештою компонентів та здійснити вилу-чення компоненти з РБС.
   8.2. Для варіантів 257-320 задіяти стратегію 2 на основі прийняття рішення рештою компонентів РБС.
   8.3. Для варіантів 321-340 задіяти стратегію 3 на основі прийняття рішення рештою компонентів.
   8.4. Для варіантів 341-484 задіяти стратегію 4 на основі прийняття рішення всіма компонентами з РБС та обміну інформацією між ними.
   8.5. Для варіантів 485-502 задіяти стратегію 5 на основі прийняття рішення всіма компонентами з РБС та обміну інформацією між ними.
   8.6. Для варіантів 503-512 задіяти стратегію 6 на основі прийняття рішення компонентою та обміну інформацією з рештою компонентів РБС.

9. Обчислення значення імовірностей у станах компонент і вимога для інших компонент здійснити обчислення ймовірності бути ураженої для всієї РБС. Цей крок здійснюється позапланово через дослідження наявного зловмисного прояву в одній з КС.
10. Здійснення оптимізації вектора, що додається в базу зловмисних дій та атак, за генетичним алгоритмом.
11. Формування імовірностей перебування в станах для надсилання іншим компонентам для визначення стану РБС (за формулами 1 та 3).
12. Залучення засобів для забезпечення стійкості компонент у КС при групі подій з кроку 8 (табл. 1), які відносяться до зовнішніх впливів.

Таким чином, згідно з розробленим методом, компоненти РБС навчені і мають змогу досягати таких цілей: вилучення ймовірно уражених компонент з РБС, встановлення відношення до ЗПЗ типу бот-мереж на основі обміну і обробки знань, створення нового класу бот-мереж на основі фрагмента програмного коду. Розбудовані компоненти РБС згідно з методом такої обробки ймовірно зловмисних подій приймають рішення про зміну структури РБС та визначають уражену КС завдяки побудованій архітектурі РБС та організації взаємозв’язку її компонентів.

Розроблений метод дозволяє здійснювати виявлення бот-мереж у комп’ютерних системах локальних мереж на основі активного моніторингу системних подій та здійснення узгодження компонент РБС при прийнятті рішення.

Метод взаємодії компонентів розподіленої багаторівневої системи виявлення ЗПЗ на основі самоорганізації

Метод взаємодії компонентів РБС на основі самоорганізації та децентралізації виявлення ЗПЗ встановлює порядок здійснення комунікації між компонентами системи та обміну знаннями між ними. Він застосовуватиметься для вирішення задач верхнього рівня організації взаємодії, тобто тільки для організації взаємодії частин системи [11] і представлення її цілісною. Для вирішення проблеми з безпосереднього виявлення ЗПЗ у локальних обчислювальних мережах застосовуватимуться методи, які належатимуть до нижчого рівня системи, що включатимуть архітектурні особливості розподіленої системи і технології виявлення ЗПЗ.

Подальші процеси, що протікатимуть у мережі, які пов’язані з функціонуванням системи, задамо такими кроками методу взаємодії компонент:

1. Визначення станів компонент.
2. Обробка відповідей від компонент КС на відправлені пакети.
3. Обробка компонентою РБС невизначеностей, пов’язаних з відсутністю відповідей на відправлені пакети.
4. Сканування заданого порту КС.
5. Оцінка стану компоненти та її звірка між рештою компонент РБС на етапі обміну повідомленнями.
6. Визначення стану РБС за формулами 1 та 3.
7. Прийняття рішення про подальшу роботу РБС, у цілому, на основі дослідження її стану компонентами за формулою 2.
8. Вилучення активної компоненти з РБС у результаті вимкнення КС.
9. Події, які активують методи виявлення ЗПЗ, впливають на зміну стану компоненти РБС; здійснення дослідження інших КС на наявність подібних активностей та обмін отриманими результатами.
10. Обробка та оптимізація статистичних даних, накопичених у системі кожною компонентою окремо.
11. Обмін знаннями в середині РБС.
12. Сумісне виконання завдань компонентами РБС.
13. Робота РБС у складі всього однієї компоненти.
14. Поповнення РБС новими компонентами.

Для визначення стану безпеки РБС використаємо дані в поточний момент часу з її компонент: стан кожної компо-

ненти від початку поточного запуску, часу перебування в кожному стані кожної компоненти, рівні безпеки в кожному стані кожної компоненти. Обчислення за формулами (1) і (2) стану РБС кожною компонентою на основі отриманих даних зі всіх активних компонент РБС здійснюємо в два етапи. На першому етапі рівень безпеки РБС визначимо за формулою 1:

$$R_{b,РБС,1} = \frac{\sum_{l=1}^{n}(1 - \sum_{s=1}^{m} k_{s,l} * p_{s,l})}{n}, \qquad (1)$$

де $R_{b,РБС,1}$ – рівень безпеки РБС, визначений на першому етапі, b – позначення безпеки, l – номер компоненти РБС, n – кількість компонент РБС, $k_{s,l}$ – коефіцієнт загрози бути ураженим ЗПЗ s – того стану компоненти, значення якого встановлюється з відрізку $[0;1]$ залежно від того, які функціональні навантаження закладено у певний s–ий стан, $p_{s,l}$ – імовірність бути ураженим ЗПЗ, m – кількість станів компонент.

За формулою 2 РБС здійснює визначення свого центру в поточний момент, а також на основі цього значення здійснюється виділення критичних компонент.

$$g(R_{b,РБС,1}, k, s, s_{c,РБС}) = \begin{cases} 0, \text{якщо виконується умова } 1 \\ 1, \text{якщо виконується умова } 2 \\ 2, \text{якщо виконується умова } 3 \end{cases}, \qquad (2)$$

де $g(R_{b,РБС,1}, k, s, s_{c,РБС})$ – функція визначення подальших кроків для РБС, $R_{b,РБС,1}$ – рівень безпеки РБС, який отримано на першому етапі за формулою 1, k – кількість активних компонент із загальної кількості n, s – номер стану, $s = 1,2,...,m$, m – кількість станів компонент, $S_{c},_{РБС}$ – середнє значення для РБС на основі сукупності станів її компонент. Умови для задання функції g представимо в таблиці 3.

Таблиця 3. Значення функції $g(R_{b,РБС,1}, k, s, s_{c,РБС})$

Умови для функції g	Значення умов для функції g		
	Значення рівня безпеки $РБС, R_{b,РБС}$	Умови, які пов'язані з кількістю ПМ	Умови, які пов'язані з середньоквадратичним відхиленням
Умова 1	$R_{b,РБС} > 0,75$	$\frac{k}{n} > 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 1$ або для $S = 8$
	$R_{b,РБС} > 0,75$	$\frac{k}{n} > 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 2$ або для $S = 3$ або $S = 4$
	$R_{b,РБС} > 0,75$	$\frac{k}{n} < 0,5$	$\min(\frac{S_{cРБС}}{S} - 1) < 1$ для $S = 5$ або для $S = 6$ або $S = 7$
	$0,5 < R_{b,РБС} < 0,75$	$\frac{k}{n} < 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 2$ або для $S = 3$
Умова 2	$0,5 < R_{b,РБС} < 0,75$	$\frac{k}{n} > 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 2$ або для $S = 3$
	$0,5 < R_{b,РБС} < 0,75$	$\frac{k}{n} < 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 2$ або для $S = 3$ або $S = 4$
	$0,25 < R_{b,РБС} < 0,5$	$\frac{k}{n} > 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 1$ або для $S = 8$
Умова 3	$R_{b,РБС} > 0,75$	$\frac{k}{n} > 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 6$ або для $S = 7$
	$0,5 < R_{b,РБС} < 0,75$	$\frac{k}{n} > 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 5$ або для $S = 6$ або $S = 7$
	$0,25 < R_{b,РБС} < 0,5$	$\frac{k}{n} > 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 4$ або для $S = 5$ або для $S = 6$ або для $S = 7$
	$0,25 < R_{b,РБС} < 0,5$	$\frac{k}{n} < 0,5$	$\min(\frac{S_{c},_{РБС}}{S} - 1) < 1$ для $S = 5$ або для $S = 6$ або $S = 7$
	$R_{b,РБС} < 0,25$	-	-
	решта випадків		

Загальна кількість таких випадків може бути 64, бо є чотири випадки для рівня безпеки, два випадки для кількості компонент, які входять до центру РБС у поточний момент часу, вісім – для віднесення центру до одного зі станів за рахунок дослідження його відхилення.

При виконанні умови 1, тобто, якщо $g(R_{b,РБС,1}, k, s, s_{c,РБС}) = 0$, то РБС продовжує роботу в режимі, коли її компоненти працюють в тих станах, в яких були. При цьому жодних дій по обробці ситуацій в певних відібраних КС не проводиться.

При виконанні умови 2, тобто, якщо $g(R_{b,РБС,1}, k, s, s_{c,РБС}) = 1$, то РБС продовжує роботу в режимі, коли її компоненти працюють у тих станах, у яких були. А також РБС зразу відмічає компоненти, для яких потрібне додаткове уточнення стосовно задач, які виконують у поточний момент часу.

При виконанні умови 3, тобто, якщо $g(R_{b,РБС,1}, k, s, s_{c,РБС}) = 2$, то РБС переходить до другого етапу уточнення свого стану на основі залучення часових характеристик станів усіх компонент.

Якщо ймовірність бути ураженим ЗПЗ впливатимуть не тільки на компоненти або їх вплив на ці модулі несуттєвий, то це не дозволяє визначити стан РБС як критичний. Такий випадок можливий, коли дослідження на першому етапі через визначення середнього значення і його подальшого використання було невисоким через невеликий час роботи від останнього запуску ПЗ компоненти чи через необхідність його усереднення на вісім станів. Але може виявитись, що багато компонент тривалий час перебувають чи перебували в одному і тому ж стані, а використання критеріїв першого етапу їх не виділяє. Тому, щоб урахувати такі граничні особливості, виділимо ймовірності бути ураженим ЗПЗ для РБС у певних визначених станах і здійснимо оцінку таких випадків на другому етапі дослідження. Для другого етапу визначення стану РБС узагальнена формула 3 для визначення рівня безпеки:

$$R_{b,РБС,2} = \frac{1}{4} * \Bigg(\sum_{s=1}^{m} \Big(1 - \prod_{\substack{j=1,\\ p_{s,j}<1}}^{n} (1 - p_{s,j})\Big) * k_s + \sum_{j=1}^{n} \sum_{\substack{s=1,\\ t_{s,j}>0\\ w_{s,j}>0}}^{m} \Bigg(\frac{t_{s,j}}{\sum_{s=1}^{m} t_{s,j}} * \frac{w_{s,j}}{\sum_{s=1}^{m} w_{s,j}} \Bigg) + \sum_{s=1}^{m} \Bigg((1 + k_s) * \frac{\sum_{j=1}^{n} w_{s,j}}{\sum_{s=1}^{m} \sum_{j=1}^{n} w_{s,j}} * \frac{\sum_{j=1}^{n} t_{s,j}}{\sum_{s=1}^{m} \sum_{j=1}^{n} t_{s,j}} \Bigg) \Bigg), \quad (3)$$

де $R_{b,РБС,2}$ – рівень безпеки РБС, визначений на другому етапі, b – позначення без-пеки, s – номер компоненти РБС, n – кількість компонент РБС, m – кількість станів компоненти, k_s – коефіцієнт загрози бути ураженим ЗПЗ s – того стану компоненти, значення якого встановлюється з відрізку $[0; 1]$ залежно від того, які функціональні навантаження закладено у певний s–ий стан, $p_{s,j}$ – імовірність бути ураженим ЗПЗ, $w_{s,j}$ – кількість перебувань компоненти з номером j у стані s, $i = 1,2,...,n$, $s = 1,2,...,m$, $t_{s,g}$ – сумарний час перебування компоненти з номером j у стані sn – кількість компонент РБС. Значення $p_{s,j}$ отримуються на основі результатів функціонування закладених у компоненти підсистем виявлення певних типів ЗПЗ.

Таким чином, згідно з розробленим методом взаємодії компонент РБС, на основі самоорганізації підтримується цілісність такої децентралізованої системи та гнучке переналаштування на подальше виконання завдань.

Експерименти

Метою експериментів була перевірка застосування методу виявлення бот-мереж, роботи класифікатора в структурі розподіленої системи та визначення залежності відсотка виявлених вузлів бот-

мережі від їх представлення векторами. Для проведення експериментів було здійснено конструювання 28 штучних бот-мереж та отриманих кодів відомих виявлених бот-мереж, згруповано їх за класами, виділено в них 25 структурних елементів у трьох стадіях функціонування і 81 функцію, причому не всі так отримані бот-мережі містили повністю всі структурні елементи та функції. Кожну функцію задано векторами зловмисних дій та атак, з врахуванням варіацій, і на їх основі побудовано зразки для включення їх у підкласи та класи. Експеримент проводився для класифікатора без додавання екземплярів створених бот-мереж та з ними, тобто здійснювалась перевірка без навчання класифікатора на створених зразках і з попереднім віднесенням зразків по класах. Другий варіант є необхідним для перевірки точності віднесення до класів тих зразків, які в них введені, бо при здійсненні моніторингу API-функцій можуть бути похибки. А також, для встановлення величини різниці у двох випадках без попереднього навчання і з ним. Це необхідно, щоб перевірити залежність виявлення за векторами по кожному класу і загальну кількість виявлених вузлів бот-мереж та точність виявлення класифікатором. Тривалість моніторингу КС локальної мережі становила 96 годин для кожного екземпляра бот-мережі кожного з двох класифікаторів. Атака з вузлів бот-мережі не здійснювалась. Вузли бот-мережі працювали тільки в режимі контролю КС та підтримки структури бот-мережі через відправлені повідомлення. Таким чином, для компонент РБС об'єктами дослідження були запущені в КС процеси і, відповідно, побудова векторів по них. Для проведення експерименту були обрані бот-мережі, які використовують стратегію отримання повного контролю в КС. Для здійснення експерименту засобами API моніторингу в КС було отримано вектори, які включали почергово оброблені класифікатором компоненти. Результати обробки представлено в табл. 4.

Експерименти передбачали визначення наступних показників ефективності виявлення вузлів бот-мереж для класів і підкласів Баєсівського класифікатора:

1) $P_{1,1}$ – відсоток векторів зловмисних дій та атак для вузлів бот-мереж, що належать даному класу відносно всіх тестових зразків, які система віднесла до цього класу з використанням попереднього навчання;
2) $P_{1,2}$ – аналогічно до 1) тільки без використання попереднього навчання;
3) $P_{2,1}$ – відсоток векторів зловмисних дій та атак для вузлів бот-мереж, що належать даному підкласу класу відносно всіх тестових векторів, які система віднесла до цього підкласу класу в тестовій вибірці (ті, які були правильно віднесені до підкласів) з використанням попереднього навчання;
4) $P_{2,2}$ – аналогічно до 3) тільки без використання попереднього навчання;
5) $P_{3,1}$ – відсоток правильно виявлених вузлів бот-мереж з використанням попереднього навчання;
6) $P_{3,2}$ – аналогічно до 5) тільки без використання попереднього навчання;
7) $P_{4,1}$ – відсоток хибно класифікованих вузлів бот-мереж як корисних додатків (помилка 1-го роду) з використанням попереднього навчання;
8) $P_{4,2}$ – аналогічно до 7) тільки без використання попереднього навчання;
9) $P_{5,1}$ – відсоток неправильно класифікованих вузлів бот-мереж як таких, що є вузлами бот-мереж, але віднесені не до того класу (помилка 3-го роду) з використанням попереднього навчання;
10) $P_{5,2}$ – аналогічно до 9) тільки без використання попереднього навчання.

Результати оцінки ефективності виявлення програмного забезпечення вузлів бот-мереж на основі роботи двох класифікаторів для введених класів та підкласів у класифікаторі наведено у табл. 4.

Таблиця 4. Результати експерименту

Показники експерименту	Отримані значення для різних класів							Середні значення
	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	
$P_{1,1}$, %	90,74	84,29	73,66	86,30	94,04	94,18	96,60	89,44
$P_{1,2}$, %	75,93	63,57	60,22	70,32	68,77	67,60	69,36	67,71
$\| P_{1,1} - P_{1,2} \|$, %	14,81	20,72	13,44	15,98	25,27	26,58	27,24	21,73
$P_{2,1}$, %	85,80	83,57	72,58	85,39	98,88	93,92	96,60	88,42
$P_{2,2}$, %	74,69	63,57	59,14	70,32	67,37	66,58	67,66	66,80
$\| P_{2,1} - P_{2,2} \|$, %	11,11	20	13,44	15,07	31,57	27,34	28,94	21,62
$P_{3,1}$, %	92,11	84,21	71,93	89,47	90,53	88,42	93,68	87,72
$P_{3,2}$, %	76,32	57,89	63,16	64,91	71,58	54,74	75,79	65,89
$\| P_{3,1} - P_{3,2} \|$, %	15,79	26,32	8,77	24,56	18,95	33,68	17,89	21,83
$P_{4,1}$,%	7,89	14,47	28,07	10,53	7,37	11,58	6,32	11,70
$P_{4,2}$,%	21,05	40,79	36,84	31,58	24,21	44,21	22,11	31,97
$\| P_{4,1} - P_{4,2} \|$,%	13,16	26,32	8,77	21,05	16,84	32,63	15,79	20,27
$P_{5,1}$,%	0	1,32	0	0	2,11	0	0	0,01
$P_{5,2}$,%	2,63	1,32	0	3,51	4,21	1,05	2,11	2,14
$\| P_{5,1} - P_{5,2} \|$, %	2,63	0	0	3,51	2,1	1,05	2,11	2,13

У результаті проведення експерименту отримано віднесення до потрібного підкласу та класу, отриманих на основі моніторингу векторів з точністю до 66% для класифікатора без введених векторів 28 штучно зґенерованих бот-мереж та 88% для класифікатора, у який попередньо було додано вектори шляхом здійснення його навчання, зберігаючи в ньому шаблони попередніх наповнень. Перевірка здійснювалась окремо для класів, їх підкласів та, в цілому, для вузлів. Результати були усереднені і їх дисперсія відносно середнього значення становить 1%. Різниця відхилення для двох класифікаторів по кожному класу, підкласу і вузлах бот-мереж та в цілому складає 21,5%. Відхилення між різницями відхилень для двох класифікаторів складає по кожному окремому класу, підкласу та вузлу бот-мережі менше 5%, що вказує на точність визначення в різних класах і підкласах. Це означає, що результат виявлення програмного забезпечення вузлів бот-мереж співпадає в розрізі класів та підкласів для векторів зловмисних дій та атак.

Помилки 1-го роду склали для першого і другого класифікаторів 11,7% та 31,97%, що пояснюється їх різним наповненням. Помилки 3-го роду – 0,01% та 2,14% відповідно, що пояснюється більш широким полем класифікації другого класифікатора через менший обсяг навчальної вибірки. У цілому, результати роботи класифікаторів показують можливість їх застосування для задач виявлення бот-мереж.

Висновки

У статті представлено метод виявлення бот-мереж розподіленими системами на основі самоорганізації та метод взаємодії її компонентів. Особливістю розроблених методів, які підтримують самоорганізацію розподіленої системи, є самостійність у прийнятті рішень компонентами системи без стороннього впливу, організація взаємодії компонентів РБС

для визначення стратегії подальшої роботи, прийняття рішень про кількість компонентів системи динамічно, прийняття рішень про наявність ЗПЗ у певних КС через залучення інших компонентів РБС з наступним відключенням таких КС, обробка і передача знань про нове виявлене ЗПЗ від одних компонентів системи іншим, оцінка стану безпеки окремих компонентів РБС та її в цілому.

Метод виявлення бот-мереж складається з двох частин: хостового і мережного рівнів. На рівні хостової частини процедура виявлення базується на реалізації класифікації Баєса, причому може бути використаний і інший класифікатор. Мережний рівень розширює результати, отримані на рівні хоста, до решти локальної мережі. Розроблений метод забезпечує обмін результатами, отриманими за класифікацією Баєса, для подальшого використання іншими компонентами розподіленої самоорганізованої системи. Результати експерименту показали, що точність виявлення бот-мереж досягає 88%.

Для підтримки цілісності РБС розроблено метод взаємодії її компонентів, який на основі її самоорганізації визначає подальшу її стратегію роботи.

Напрямками подальших досліджень є розробка нових методів виявлення ЗПЗ для наповнення РБС. Розроблені методи повинні орієнтуватись на особливості архітектури розподіленої самоорганізованої системи та використовувати цю перевагу над іншими хостовими методами.

References

1. Security Response Publications (2019). Monthly Threat Report. [Online] Available: https://www.symantec. com/security_response/publications/monthlythreatreport.jsp.
2. McAfee Labs (2019). McAfee Labs Threat Report. December 2017. [Online] Available: https://www.mcafee.com /us/resources/reports/rp-quarterly-threats-dec-2017.pdf.
3. Symantec (2019). Overview of Symantec Endpoint Protection 12. Part 2. [Online] Available: https://www.anti-malware.ru/reviews/Symantec_Endpoint_Protection_12_2.
4. Palo Alto Networks (2019). [Online] Available: https://www.paloaltonetworks.com/
5. Malwarebytes. (2019). Malwarebytes Endpoint Security [Online] Available: https://ru.malware bytes.com/business/endpointsecurity/
6. Cisco (2019). Cisco NAC Appliance (Clean Access). [Online] Available: https://www.cisco. com/c/en/us/products/security/nac-appliance-clean-access/index.html.
7. Comodo (2019). ComodoCyberSecurity. [Online] Available: https://www.comodo.com/
8. Kumar, N.J., Singh, P., Bali, R.S., Misra, S., Ullah, S. (2015). An intelligent clustering scheme for distributed intrusion detection in vehicular cloud computing, Cluster Computing, 18(3), 1263–1683. DOI: 10.1007/s10586-015-0463-7
9. Boukhlouf, D., Kazar, O., Kahloul, L. (2016). Network Security: Distributed Intrusion Detection System using Mobile Agent Technology, International Journal of Communication Networks and Distributed Systems, 16(4). DOI: 10.1504/IJCNDS.2016.10001612
10. Boukhlouf, D., Kazar, O. (2012). Hybrid Approach based Mobile Agent for Distributed Intrusion Detection System, Journal of Information Security Research, 3(1), 30–40. DOI: 10.1109/ICEELI.2012.6360647
11. Markowsky, G., Savenko, O., Sachenko, A. (2019). Distributed Malware Detection System Based on Decentralized Architecture in Local Area Networks. Advances in Intelligent Systems and Computing III, 871, 582–598. DOI: 10.1007/978-3-030-01069-0_42
12. Branitskiy, A., Kotenko, I. (2017). Hybridization of computational intelligence methods for attack detection in computer networks. Journal of Computational Science, 23, 145–156. DOI: 10.1016/j.jocs.2016.07.010
13. Pronoza, A., Vitkova, L., Chechulin, A., Kotenko, I. (2019). Visual Analysis of Information Dissemination Channels in Social Network for Protection Against Inappropriate Content. In Proceedings of the Third International Scientific Conference: Intelligent Information Technologies for Industry, Volume 2, Sochi, Russia, 2019 (pp.95–105). DOI: 10.1007/978-3-030-01821-4_11
14. Bezobrazov, S., Sachenko, A., Komar, M., Rubanau, V. (2016). The methods of artificial intelligence for malicious applications detection in Android OS. International Journal of Computing, 15 (3), 184–190.
15. David, B., Filiol, E., Gallienne, K. (2017). Structural analysis of binary executable headers for malware detection optimization. Journal of Computer Virology and Hacking Techniques, 13 (2), 87–93. DOI: 10.1007/s11416-016-0274-2
16. Eslahi, M., Abidin, W. Z., Naseri, M. V. (2017). Correlation-based HTTP Botnet detection using network communication histogram analysis. In Proceedings of 2017 IEEE Conference on Application, Information and Network Security, Miri, Malaysia, 2017 (pp. 7–12). DOI: 10.1109/AINS.2017.8270416
17. Sun, M., Xu, G., Zhang, J., Kim, D. (2017). Tracking you through DNS traffic: Linking user

sessions by clustering with Dirichlet mixture model. In Proceedings of 20th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, Miami, FL, US, 2017 (pp. 303–310). DOI: 10.1145/3127540.3127567

18. Schomp, K., Rabinovich, M., Allman, M. (2016). Towards a model of DNS client behavior. In Proceedings of the International Conference on Passive and Active Network Measurement, volume 9631, Heraklion, Crete, Greece, 2016 (pp. 263–275). DOI: 10.1007/978-3-319-30505-9_20
19. Zheng, J., Li, Q., Gu, G., Cao, J., Yau, D. KY, Wu, J. (2018). RealtimeDDoS Defense Using COTS SDN Switches via Adaptive Correlation Analysis. IEEE Transactions on Information Forensics and Security, 13(7), 1838–1853. DOI: 10.1109/TIFS.2018.280560

RESUME

O.S. Savenko

The Method Of Detection Of Bot-Net On Based Distributed And Self-Organization Systems

In order to detect malware in computer systems of local networks, it is suggested to use distributed detection systems. Such systems must be decentralized and self-organized in order to increase the reliability of the detection and reliability of the work. Such systems are filled with malicious software detection methods implemented in them. The paper presents the developed method of detecting botnets by distributed systems on the basis of self-organization and the method of interaction of its components. The developed methods support the self-organization of the distributed system, the autonomy in decision making by the components of the system without third-party influence, the organization of the interaction of the components to determine the strategy of further work, decision-making on the number of components of the system, making decisions about the presence of malware in certain computer systems through the involvement of other components distributed system with the subsequent disconnection of such computer systems, processing and transferring knowledge about the new detected malice not software from one component of the system to another, an assessment of the security status of individual components of the system and its overall.

The method of detecting botnets consists of two parts: the host and network levels. At the host level, the detection procedure is based on the implementation of Baeus's classification. The network layer extends the results obtained at the host level to the rest of the local network. The results of the experiment on the use of the developed system and methods showed that the accuracy of detection of botnets reaches 88%.

To maintain the integrity of the distributed self-organized system, a method of interaction of its components is developed, which, based on its self-organization, determines its further strategy of work.

Надійшла до редакції 10.12.2018